

What is claimed is:

1. A method of creating an online point-in-time view of an application table of an active database having at least one before image table and at least one after image table, the method comprising:

retrieving a retention time of the application table;

determining that a point-in-time is greater to or equal than the retention time; and

creating the point-in-time view of the application table, the view comprising a union of a set of rows from the after image table having a timestamp less than or equal to the point-in-time and a set of rows from the before image table having a timestamp less than or equal to the point-in-time and having a changing timestamp greater than the point-in-time.

2. A method of creating an online point-in-time view of an application table of an active database having at least one before image table and at least one after image table, the method comprising:

retrieving a transaction id set associated with a transaction id that begins before or at a point-in-time and ends after the point-in-time;

retrieving a retention time of the application table;

determining that the point-in-time is greater than or equal to the retention time; and

creating the point-in-time view of the application table, the view comprising a union of a set of rows from the after image table having a timestamp that is less than or equal to the point-in-time and having a transaction id that is not a proper subset of the transaction id set, and a set of rows from the before image table having a timestamp less than or equal to the point-in-time, and having a transaction id that is not a proper subset of the transaction id set, and having a changing transaction id that is a proper subset of the transaction id set or having a changing timestamp that is greater than the point-in-time.

3. A method of rollbacking at least one table of an active database to a point-in-time, having at least one before image table and at least one after image table, the method comprising:

- retrieving a retention time for each of the at least one table;

- determining that none of the retention times for the at least one table is greater than the point-in-time;

- locking all tables to be rollbacked and disabling all related integrity constraints;

- deleting rows from each of the at least one after image table having a timestamp greater than the point-in-time;

- inserting into each of the at least one after image table, rows from the associated before image table having a timestamp less than or equal to the point-in-time and having a changing timestamp greater than the point-in-time; and

- enabling all related integrity constraints.

4. A method of rollbacking at least one table of an active database to a point-in-time, having at least one before image table and at least one after image table, the method comprising:

- retrieving a retention time for each of the at least one table;

- determining that none of the retention times for each of the at least one table is greater than the point-in-time;

- retrieving a transaction id set associated with a transaction id that begins before or at the point-in-time and ends after the point-in-time;

- locking all tables to be rollbacked and disabling all related integrity constraints;

- deleting rows from each of the at least one after image table having a timestamp greater than the point-in-time or having a transaction id that is a proper subset of the transaction id set;

- inserting into each of the at least one after image table, rows from the associated before image table having a timestamp less than or equal to the point-

in-time and having a transaction id that is not a proper subset of the transaction id set, and having a changing transaction id that is a proper subset of the transaction id set or having a changing timestamp that is greater than the point-in-time; and enabling all related integrity constraints.

5. A method of rollbacking selected rows in a table of an active database to a point-in-time, the database having at least one before image table and at least one after image table, the method comprising:

retrieving a retention time of the table;

determining that the point-in-time is greater than or equal to the retention time;

locking the table to be rollbacked and disabling all related integrity constraints;

deleting selected rows from the after image table having a timestamp greater than the point-in-time;

inserting into the after image table, selected rows from the before image table having a timestamp less than or equal to the point-in-time and having a changing timestamp that is greater than the point-in-time; and

enabling all related integrity constraints.

6. A method of rollbacking selected rows in a table of an active database to a point-in-time, the database having at least one before image table and at least one after image table, the method comprising:

retrieving a transaction id set associated with a transaction id that begins before or at the point-in-time and ends after the point-in-time;

retrieving a retention time of the table;

determining that the point-in-time time is greater than or equal to the retention time;

locking the table to be rollbacked and disabling all related integrity constraints;

deleting selected rows from the after image table having a timestamp greater than the point-in-time or having a transaction id that is a member of the transaction id set;

inserting into the after image table, selected rows from the before image table having a timestamp that is less than or equal to the point-in-time and having a transaction id that is not a proper subset of the transaction id set, and having a changing transaction id that is a proper subset of the transaction id set or having a changing timestamp that is greater than the point-in-time; and

enabling all related integrity constraints.

7. A method of obtaining a history of a single data row of a table of an active database, the database having at least one before image table and at least one after image table, the method comprising:

retrieving a data row set comprising the union of zero or one selected row from the after image table and zero or more selected deleted rows from the before image table;

determining that the data row set is not empty; and

recursively finding all associated parent rows for each data row in the data row set.

8. The method according to claim 7, wherein the finding comprises recursively finding all rows having a changing timestamp that equals a previous row's timestamp, the previous row being associated with the rows.

9. The method according to claim 7, wherein the finding comprises recursively finding all rows having a changing timestamp and a changing transaction id that equals a previous row's timestamp and transaction id, the previous row being associated with the rows.

10. A method of rollbacking a transaction id set in a group of tables of an active database, the database having at least one before image table and at least one after image table, the method comprising:

retrieving an earliest starting time of the transaction id set;

- retrieving a latest retention time for all of the tables;
- determining that the earliest starting time is greater than the latest retention time;
- locking all tables to be rolled back and disabling all related integrity constraints;
- rollbacking the transaction id set in the group of tables; and
- enabling all related integrity constraints.

11. The method according to claim 10, further comprising:

- deleting rows from the after image table having a transaction id that is a member of the transaction id set;

- selecting earliest before image rows from the before image table having a transaction id that is a proper subset of the transaction id set, or having a changing transaction id that is a proper subset of the transaction id set, having an earliest timestamp;

- deleting any after image of a row from the after image table when a transaction id of the row is a proper subset of the transaction id set and a changing transaction id of the row is not equal to a current transaction id; and

- deleting any after image of a row from the after image table and inserting a before image from the row into the after image table when the transaction id of the row is not a proper subset of the transaction id set and the changing transaction id is a proper subset of the transaction id set.

12. The method according to claim 10, further comprising:

- deleting rows from the after image table having a transaction id that is a member of the transaction id set;

- selecting before image rows and any descendants from the before image table having a transaction id that is a proper subset of the transaction id set or having a changing transaction id that is a proper subset of the transaction id set;

- grouping the selected rows into families;

removing any families having an earliest transaction id that is a proper subset of the transaction id set;

inserting an earliest returned before image in a family into the after image table when the last changing transaction id of the family is equal to a current transaction id; and

inserting an earliest returned before image in a family into the after image table when the last changing transaction id of the family is a proper subset of the transaction id set and if no after image of the family exists.

13. A method of rollbacking all transactions made in a user session of an active database, the method comprising:

retrieving selected parameters associated with the user session;

retrieving names of the tables used by a user and transaction ids made by the user during a session, the session being defined by a user login time and logout time;

retrieving an earliest starting time of the transaction id set;

retrieving a latest retention time for all of the tables;

determining that the earliest starting time is greater than the latest retention time; and

rollbacking the transaction set from the tables.

14. A system for use with an active database for performing at least one of online selectively rollbacking at least one application table or selected application data in at least one table, creating online point-in-time views of application tables, providing online history images associated with the database, reconstructing equivalent SQL statements of a committed transaction, reconstructing equivalent SQL statements of a user session, and providing a selective audit trail report on demand, the system comprising:

a plurality of user application after image tables, wherein each of the tables has one before image table to store before images, and one after image view;

a table comprising a table name field, a retention time field, and an export timestamp field;

an image manager that creates image views; and

a rollback manager that manages the first table.

15. The system according to claim 14, further comprising a second table including fields associated with a user session.

16. The system according to claim 14, further comprising a third table including fields associated with a transaction id, starting timestamp, and ending timestamp.

17. The system according to claim 14, the image manager further comprising:

an inserting trigger that sets parameters for an inserted row in the after image tables;

an updating trigger that inserts an original row into a before image table and sets parameters associated with the changed row in the after image tables and the before image tables; and

a deleting trigger that inserts the original row into the before image table and sets parameters associated with the changed row in the before image tables.

18. The system according to claim 17, further comprising:
a transaction trigger that records each transaction id.

19. The system according to claim 17, further comprising:
a login/logout trigger that records a time when a user enters or exits the database.

20. The system according to claim 14, the after image tables further comprising at least one of a timestamp field, user id field, and transaction id field.

21. The system according to claim 14, wherein the rollback manager performs at least one of rollbacking at least one table to a point-in-time, rollbacking selected rows in a single table to a point-in-time, rollbacking a transaction id set in a group of tables, rollbacking all transactions made in a user session, reconstructing equivalent SQL statements of a transaction id set in a

group of tables, reconstructing equivalent SQL statements of a user session in a group of tables, and providing a selective audit trail report on demand.

22. A computer readable medium containing instructions for implementing application tables in a database and tracking images, wherein the instructions comprise executable instructions for implementing:

at least one after image table that stores current database images; and

a before image table associated with each of the at least one after image table that stores previous images of the current database images,

wherein the at least one after image table and the before image table are available for online viewing.

23. The computer readable medium according to claim 22, wherein each of the at least one after image tables further comprise at least one of a timestamp field, a userid field, and a transaction id field.

24. The computer readable medium according to claim 22, wherein each of the before image tables further comprise at least one of a changing timestamp field, a changing userid field, a changing transaction id field, and an opcode.

25. The computer readable medium according to claim 22, wherein each of the before image tables further comprise a field associated with a field in the associated after image table.

26. A method of reconstructing equivalent SQL statements of a transaction id set in a group of tables of an active database, the database having at least one before image table and at least one after image table, the method comprising:

retrieving an earliest starting time of the transaction id set;

retrieving a latest retention time for all related tables;

determining that the earliest starting time is greater than the latest retention time;

creating a delete SQL statement for each deleted row in each of the before image tables having a changing transaction id that is a proper subset of the transaction id set;

creating an update SQL statement using the after image in one of the after image table or the before image table associated thereof by the relation that the timestamp and transaction id of an associated child row are equal to a changing timestamp and changing transaction id of an associated parent row, respectively, for each updated row in each of the before image tables having a changing transaction id that is a proper subset of the transaction id set;

creating an insert SQL statement for each row in each of the before image tables having a transaction id that is a proper subset of the transaction id set and an associated timestamp and transaction id do not exist in the before image as the changing timestamp, changing transaction id; and

creating an insert SQL statement for each row in each of the after image tables having a transaction id that is a proper subset of the transaction id set and an associated timestamp and transaction id do not exist in the before image as the changing timestamp, changing transaction id.

27. A method of reconstructing equivalent SQL statements of a transaction id set in a group of tables of an active database, the database having at least one before image table and at least one after image table, the method comprising:

retrieving an earliest starting time of the transaction id set;

retrieving a latest retention time for all related tables;

determining that the earliest starting time is greater than the latest retention time;

creating a delete SQL statement for each deleted row in each of the before image tables having a changing transaction id that is a proper subset of the transaction id set;

creating an update SQL statement using an after image of an associated child row in one of the after image table or the before image table for each updated row in each of the before image tables having a changing transaction id that is a proper subset of the transaction id set; and

creating an insert SQL statement for each row without an associated parent row in each of the before and after image tables having a transaction id that is a proper subset of the transaction id set.

28. A method of reconstructing equivalent SQL statements of all transactions made in a user session of an active database, the method comprising:

retrieving selected parameters associated with the user session;

retrieving names of tables used by a user and transaction ids made by the user during the session, the session being defined by a user login time and a user logout time;

retrieving an earliest starting time of the transaction id set;

retrieving a latest retention time for all of the tables;

determining that the earliest starting time is greater than the latest retention time; and

reconstructing equivalent SQL statements of the transaction id set from the tables.

29. A method of reconstructing equivalent SQL statements of all transactions made in a user session of an active database, the database having at least one before image table and at least one after image table, the method comprising:

retrieving selected parameters associated with the user session;

retrieving names of tables used by a user during the session, the session being defined by a user login time and a user logout time;

retrieving a latest retention time for all of the tables;

determining that the login time is greater than the latest retention time;

creating a delete SQL statement for each deleted row in the before image tables having a changing userid containing a user session id and a changing timestamp between the session starting timestamp and the session ending timestamp;

creating an update SQL statement using an after image of an associated child row in one of the after image tables or the before image tables for each updated row in the before image tables having a changing userid containing the user session id and a changing timestamp between the session starting timestamp and the session ending timestamp; and

creating an insert SQL statement for each row without an associated parent row in each of the before and after image tables having a userid containing the user session id and a timestamp between the session starting timestamp and the session ending timestamp.

30. A method of creating a selective audit trail report associated with activities of an application database user during a particular period, the activities including a plurality of database sessions, the method comprising:

retrieving userids and unique session ids associated with the user during the period;

retrieving names of any tables having any transactions made during the database sessions with one of retrieved unique session ids;

retrieving an earliest start time of the sessions;

retrieving a latest retention time for all of the tables whose names were retrieved;

determining that the starting time is greater than the latest retention time;

reconstructing equivalent SQL statements for all of the sessions;

associating each reconstructed update and delete statement with an associated changing session id, changing timestamp, and changing userid, and adding the changing transaction id to the association if the database provides the user with transaction id tracking; and

associating each reconstructed insert statement with an associated session id, timestamp, and userid, and adding the transaction id to the association if the database provides the user with transaction id tracking.

31. A method of rollbacking transactions made in a user session in a group of tables of an active database, the database having at least one before image table and at least one after image table, the method comprising:

retrieving a starting time, an ending time, a unique session id of the user session;

retrieving a latest retention time for all of the tables;

determining that the starting time is greater than the latest retention time;

locking all tables to be rollbacked and disabling all related integrity constraints;

rollbacking the transactions made in the user session in the group of tables; and

enabling all related integrity constraints.

32. The method according to claim 31, wherein the unique session id comprises a unique session id provided by a database vendor.

33. The method according to claim 31, wherein the unique session id comprises a session id and the starting timestamp of the session.

34. The method according to claim 31, further comprising:

deleting rows from the after image table having a unique session id that is equal to the user session's unique session id;

selecting earliest before image rows, having an earliest timestamp, from the before image table having a unique session id that is equal to the user session's unique session id, or having a changing unique session id that is equal to the user session's unique session id;

deleting any after image of a row from the after image table if a unique session id of the row is equal to the user session's unique session id and a

changing unique session id of the row is not equal to a current session's unique session id; and

deleting any after image of a row from the after image table and inserting a before image from the row into the after image table if the unique session id of the row is not equal to the user session's unique session id set and the changing unique session id is equal to the user session's unique session id.

35. The method according to claim 30, further comprising:

deleting rows from the after image table having a unique session id that is equal to the user session's unique session id;

selecting before image rows and any descendants from the before image table having a unique session id that is equal to the user session's unique session id or having a changing unique session id that is equal to the user session's unique session id;

grouping the selected rows into families;

removing any families having an earliest unique session id that is equal to the user session's unique session id;

inserting an earliest returned before image in a family into the after image table when the last changing unique session id of the family is equal to a current session's unique session id; and

inserting an earliest returned before image in a family into the after image table when the last changing unique session id of the family is equal to the user session's unique session id and if no after image of the family exists.